

Lab Week 10 Checkoff Sheet

Multithreaded Programming (Submission deadline Friday 09/02 2:00 PM)

G09

Stage 1: (10 Points)

Following Lab Description, first clean and build `tirtos_builds` project. Then clean and build `multiplethreads` and `threadcomms` projects. Take care of any 'Dependencies' as given in the Lab Description.

Does `tirtos_builds` project build correctly without errors? ☒ YES ☐ NO (Circle the correct answer)

Does `multiple_threads` build correctly without errors? ☒ YES ☐ NO

Does `thread_comms` build correctly without errors? ☒ YES ☐ NO

If your answer to one or more of the above is NO, submit screenshot (s) of error messages (<5 points)

If all answers are YES, first run `multiplethreads`

LED0 should blink with 2 secs period. UART should merely echo the input character

Initials KSP Date 08/26 Time 9:20 (5 points)

Then run `threadcomms`

When 'F' is typed at the terminal LED1 should go OFF. When 'O' is typed at the terminal LED0 should come ON

Initials KSP Date 08/26 Time 9:20 (5 points)

Stage 2: (10 Points)

Now modify `multiplethreads` and associated .c files (`main_tirtos.c`, `blinkyThread.c` and `uartThread.c`) in such a way that a number `n` between 2 and 5 (both inclusive) typed at the terminal by `uartThread` is passed on to `blinkyThread` via a message queue and `blinkyThread` causes LED0 to flash `n` times with a period of 1 sec. You can refer to `threadcomms` project to find out how the communication between the two threads is established.

Have you completed this stage successfully? This implies that you are able to enter a single number between 2 and 5 thru the UART terminal and LED0 blinks the number of times entered and then stops blinking

Initials KSP Date 08/26 Time 9:20 (10 points)

Stage 3: (10 Points)

As per Lab Description, uartThread should be able to continuously accept inputs between 2 and 5 from the terminal and pass it on to the blinkyThread via the message queue, which implies that it is not necessary to wait for the current flashing of LED0 to stop before typing the next input at the terminal as long as there is space in the message queue i.e the queue does not exceed its maximum size. Make any modifications in your code developed in Stage 2 to make this possible.

Have you completed this stage successfully? You should now be able to enter multiple numbers between 2 and 5 while the LED is still flashing and these numbers should be placed in the message queue and the LED will blink sequentially corresponding to each number entered. For example if you enter 5,3, and 2 (all without waiting for the LED to stop blinking) the LED0 should blink a total of $5+3+2 = 10$ times

Initials KSV Date 08/26 Time 9:20 (10 points)

Instructor / TA's comments (whether the project performs as per specifications; if not, describe the deviations and partial credit given)

Works well

GRADE / 30

What to submit:

Zipped projects for Stage 1 and Stage 3 + Checkoff sheet if Stage 3 is working OR,

Zipped projects for Stage 1 and Stage 2 + Checkoff sheet if Stage 3 is not working
OR,

Zipped project for Stage 1 if neither Stage 2 nor Stage 3 is working OR,

Screenshots of errors + Checkoff sheet if even Stage 1 is not working

Absolutely no self checkoffs